IT'S A JUNGLE OUT THERE: FILTERING HIGH QUALITY REVIEWS FROM AMAZON USING ATTENTION-BASED MODELING

An Undergraduate Research Scholars Thesis

by

JONATHAN INNIS

Submitted to the Undergraduate Research Scholars Program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. James Caverlee

May 2020

Major: Computer Science

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

ABSTRACT

It's a Jungle Out There: Filtering High Quality Reviews
from Amazon Using Attention-Based Modeling

Jonathan Innis
Department of Computer Science & Engineering
Texas A&M University

Research Advisor: Dr. James Caverlee
Department of Computer Science & Engineering
Texas A&M University

Today, a large amount of misinformation proliferates on the internet, including online review websites. Because of the prevalence of fake or unknowledgeable reviewers, there is a need for new models to order and filter reviews. Hence, this thesis explores the potential for new machine learning methods trained over carefully curated "expert" reviews to automatically uncover high-quality reviews from large review collections. Concretely, we leverage machine learning and deep learning models to capture the semantic, grammatical, and argumentative structure of a high-quality review such that we are able to filter out fake or unknowledgeable reviews. We provide evidence showing that attention-based modeling is able to capture this semantic and argumentative structure such that we can produce high performance in delineating high-quality reviews from low-quality reviews. Specifically, we leverage different training methods coupled with multiple machine learning and deep learning models to identify the highest performing training method and model. We find that cross-domain training coupled with attention-based modeling on sentence-based high-quality review filtering produces the highest performance, outperforming other models and training methods by just under five percent. Additionally, we find that this model shows strong evidence of generalizing our task of high-quality review filtering outside of the initial domains on which the model was trained. Thus, we are able to show proof-of-concept that automated high-quality review filtering can now be captured with advanced modeling techniques.

ACKNOWLEDGMENTS

First, I would like to thank my research advisor, Dr. James Caverlee, for all of his encouragement and support in pursuing research in this topic. Without such an amazing research advisor, I would not have been as encouraged or as excited to pursue this particular research area. Additionally, I would like to thank everyone in Dr. Caverlee's lab, namely Ziwei Zhu, and Majid Alfifi for providing me resources to perform my research and to bring the server back up to working order when it broke in the middle of the year. Additionally, I would like to thank my parents, Jim and Kelli, and my sister, Lauren, for the support they have provided throughout my life and the encouragement they have provided to passionately pursue my interests. Finally, I would like to thank my roommates and friends who constantly give me encouragement and support and have made this culmination of my experience at Texas A&M University a special one.

NOMENCLATURE

NLP	Natural Language Processing
TFIDF	Term Frequency-Inverse Document Frequency
SVM	Support Vector Machine
DNN	Deep Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
BERT	Bidirectional Transformers
GLUE	General Language Understanding Evaluation
SQuAD	Stanford Question Answering Dataset
SWAG	Situations With Adversarial Generations
Curated Review Dataset	A dataset that contains pre-determined high-quality reviews
Uncurated Review Dataset	An ambiguous dataset that contains a combination of high-quality and low-quality reviews

CHAPTER I

INTRODUCTION

Over the last 10 years, websites like Amazon and Ebay have dramatically transformed the way that we purchase consumer products. As Amazon became a larger e-commerce marketplace during the late 1990s and into the early 2000s, more and more consumers began moving their shopping away from traditional retail stores into the online marketplace [2]. From 2017 to 2019, there has been over a thirty percent increase in retail e-commerce sales, and that trend is expected to continue by increasing the overall retail e-commerce sales from 365 million dollars in 2019 to almost 600 million dollars in 2024 [2]. Clearly, today's product market is shifting from consumers purchasing products in retail stores to purchasing products online. Because these consumers are unable to use or wear the products before purchasing them, a vast majority of people rely on user reviews to differentiate between high-quality and low-quality products. As the paper *The Impact of New Media on Customer Relationships* emphasizes, customer interactions are changing drastically as customers have the ability to interact not just with sellers but with other consumers [3]. The paper states, "consumers have become highly active partners, serving as customers as well as producers and retailers, being strongly connected with a network of other consumers" [3].

E-commerce sites often aggregate reviews to represent the overall consumer opinion of the product. In the case of Amazon, the e-commerce giant aggregates reviews by providing an overall ranking out of five stars as well as displaying the most helpful and critical reviews immediately to the user. As with the general consumer reviews, what Amazon determines to be the most helpful and most critical reviews is crowd sourced. Essentially, all consumers who don't have the time to scour through thousands of Amazon reviews for a product will look at the overall rating and, if they have time, look over the most helpful and most critical review associated with a product. The main problem with the process of simply looking at the aggregate rating associated with a review is that reviews can easily be faked or highly misleading [4, 5, 6, 7]. Just by spending a limited

amount of time on Amazon, it is not hard to find evidence of users who have purchased a product, reviewed a product, all while having no idea how to functionally use a product [4, 5, 6, 7].

While some consumers may not care how an e-commerce giant such as Amazon ranks their reviews, research shows that better review ranking produces a better consumer experience and greater customer retention [8]. The paper *What Makes a Helpful Online Review? A Study of Customer Reviews on Amazon.com* by Mudambi, et al explains, "Online retail sites with more helpful reviews offer greater potential value to customers. Providing easy access to helpful reviews can create a source of differentiation. In practice, encouraging quality customer reviews does appear to be an important component of the strategy of many online retailers" [8].

1.1 Goals

Because of the importance of the helpfulness of a rating on a user's e-commerce experience, studies such as the previously mentioned Mudambi paper look deeper into what constitutes a "helpful" consumer review [8]. Rather than looking explicitly at user-generated feedback to determine a high-quality (or helpful) review, our research suggests there is a way to automate the process of ranking consumer reviews.

In our research, we deem that researchers who understand products within a certain scope are "domain experts" and produce above-average or "high-quality" reviews. We assume that the traditional crowd-sourced model of representing the quality of a product is not accurate. Instead, we assume that because domain experts have a deeper understanding of a product and produce higher quality reviews, they should drive the conversation and perception of it. In the following section, we outline specific goals of our research centered around identifying "high-quality" reviews within an uncurated review dataset.

Our research aim is to determine if we can identify these "high-quality" reviews from an uncurated review dataset. Specifically, we wish to scrape data from a curated dataset containing high-quality reviews and data from an uncurated dataset where we have yet to determine whether these reivews are of high or low quality. By scraping these reviews and attempting to model this relationship, we aim to understand what differentiates specific uncurated reviews as high-quality

reviews versus low-quality reviews. Additionally, we aim to build a model that does not just capture the relationship between a single curated dataset domain into the same uncurated domain but captures a generalized model of high-quality reviews.

1.2 High-Quality Reviews: Curated vs. Uncurated

In this thesis, we propose to identify high-quality reviews by first identifying a collection of curated reviews and a collection of uncurated reviews. In this section, we observe the separation between the curated and uncurated dataset. Additionally, we evaluate the logic behind the choice of Wirecutter as the curated dataset and Amazon as the uncurated dataset.

Curated Review Dataset

Curated review data represents specific reviews that we identify as expert review data. This expert review data is the baseline by which we evaluate all our uncurated reviews.

We formed our curated dataset by scraping all our data from Wirecutter. While there are many websites that contain product reviews, Wirecutter was chosen as our curated dataset because of the expertise and impartiality of the reviewers [9]. As a New York Times company, Wirecutter goes through vigorous processed to assure the readers that the products they are presenting are of the highest quality and most interest to the readers [9]. Wirecutter conducts all its reviews solely based on its editorial team, without input from the finance team [9].

Uncurated Review Dataset

Uncurated review data represents specific reviews that is indeterminant on whether it is expert or non-expert review data. We wish to separate this uncurated data using a machine-learning classifier into both expert and non-expert review data.

We formed our uncurated dataset by scraping data from Amazon corresponding to a product review that we scraped from Wirecutter. Because of the scale of Amazon's e-commerce marketplace, forming our uncurated review dataset from their online review database proved the easiest. Nearly every review that was scraped from Wirecutter had a corresponding review on Amazon. Additionally, most Amazon reviews that we scraped had a relatively large number of consumer reviews (1000+).

1.3 Challenges

Next, we discuss challenges that we foresee as potential issues with the data and our research methods.

Data Parallelization

In our experimental research, we collected our curated review set through Wirecutter. Because Wirecutter has a specific set of editors and a specific way to write their product reviews, there are certain inherent biases and styles to the way that an editor writes on Wirecutter. When we are performing high-quality/low-quality classification on our uncurated dataset, we are assuming that a Wirecutter review is representative of a "high-quality" review in the uncurated dataset.

This assumption could be potentially problematic as it could introduce both false positives and false negatives into the classification of the dataset. In other words, there is potential to classify reviewers who may not fully understand a product but resemble the semantic structure of a Wirecutter reviews as an expert reviewer while missing reviewers who may understand the product but do not resemble the semantic structure of a Wirecutter reviewer.

Positive Review Overfitting

Because of the nature of our curated review dataset, the reviews appearing on Wirecutter are mostly the best products; thus, many reviewers on Wirecutter use highly positive language. Observing uncurated reviews on Amazon, we can see that not only does highly positive language exist, but some negative language also exists.

Expert review websites will often only review products that are highly touted and loved by the consumer. This is a direct consequence of these companies needing views from users and only having limited resources to review certain products.

Because expert review websites are often not willing to take the time to review products that they know are going to be poor and product negative reviews, it is extremely difficult to model the more negative review structure that exists in our uncurated review dataset.

Domain Generalization

The scope of our curated dataset is necessarily smaller than our uncurated. Because of this,

we have specifically chosen domains that we wish to learn from in our curated dataset and classify using our uncurated. While we aim to model domains in our curated dataset into our uncurated dataset, we recognize that domains may not model other domains. Thus, it may be particularly difficult to generalize this modeling procedure throughout many domains.

1.4 Contributions

In summary, this thesis makes three unique contributions:

(i) First, this research proposes a new approach to identifying high-quality reviews from an uncurated dataset through training a pre-specified high-quality dataset (such as Wirecutter).

(ii) Second, this research introduces cutting-edge attention-based modeling and BERT embeddings to capture the complex argumentative and semantic structure of high-quality reviews.

(iii) Third, this research proposes a method of generalizing the training process of high-quality review classification through cross-domain training.

We find that a cross-domain training coupled with attention-based modeling on sentence-based high-quality review filtering produces the highest performance, outperforming other models and training methods by just under five percent. Additionally, we find that this model shows strong evidence of generalizing our task of high-quality review filtering outside of the initial domains on which the model was trained. Thus, we are able to show proof-of-concept that automated high-quality review filtering can now be captured with advanced modeling techniques.

CHAPTER II

RELATED WORK

Extensive related research has been performed related to the semantic structure of online reviews. In this section, we observe the progression of research in the domain of online crowd-sourced reviews and how this work relates to our work in expert-classification of crowd-sourced reviews. In particular, because this thesis focuses on the classification of high-quality user reviews, we will look at the origins of online product review classification and summarization, the origin of low-quality user reviews through the means of online opinion spam, and modern deep learning architectures we have leveraged in our work to assist with the classification of high-quality reviews.

2.1 Review Classification & Summarization

As the landscape of online reviews has increased within the last ten years, so too has research into the classification of these reviews. Introductory research into the classification of online reviews focused most prominently on data summarization and the classification of review sentiment [10, 5, 11]. Because of the wide scope of the internet as a medium, there is currently a large breadth of online consumer review domains. Thus, the problem of sentiment classification and summarization of online reviews is still being researched and explored.

2.2 Opinion Spam

As the landscape of consumer reviews has grown, *internet spam* has grown with it. We define *internet spam* as information that resides on the internet for the strict purpose of artificially increasing the score of a product, service, or web-page. Researchers have worked extensively to gain a deeper understanding of how web spam has affected the way we perceive information in general [6, 12, 13, 14].

Opinion spam is contained within the larger domain area of *internet spam*. We define *opinion spam* as online consumer reviews that are written without prior knowledge of a product or

service and serve to artificially increase the review score of that product or service. Studies have specifically detailed machine learning and deep learning methods for classifying opinion spam in consumer reviews [4, 7, 15].

2.3 Review Quality Assessment

Though opinion spam serves as a large issue within product reviews, there also exist numerous examples of low-quality reviews for a given product on a review website. Differentiating between low-quality or fake reviews and high-quality reviews on review sites is the basis for our research. Current studies have provided experimental methods to re-classify online reviews based on semantic properties, readability analysis, etc. [16, 17, 18, 19]. These measures have proven effective as features of traditional machine learning models, but limited research has been conducted using modern deep learning architectures to map a curated set of high-quality "expert" reviews onto an uncurated set of reviews containing fake user reviews, low-quality user reviews, and high-quality user reviews combined together.

2.4 Modern Deep Learning Architectures

In this subsection, we introduce various modern-day deep learning architecture which we utilize in our research. In particular, we introduce long short-term memory classification, attention-based modeling, and BERT embeddings.

Long Short-Term Memory Classification

While more advanced forms of Recurrent Neural Networks are now used in modern deep learning-based architectures, Long Short-Term Memory (LSTM) models still serve as the foundation for modeling long-term dependencies in natural language processing. Near the end of the 1990s, a group of researchers released a paper introducing the concept of LSTM-based architectures [20]. This architecture radically increased performance from traditional machine-learning-based architectures and past deep-learning-based architectures. We utilize LSTM modeling in our high-quality review classification.

Attention-Based Modeling

Cutting-edge deep-learning architectures that better capture long-term dependencies have been utilized to capture complex natural language processing tasks. While long short-term memory was not able to capture some dependencies if input was too long, attention-based modeling attempted to remedy this problem [21]. The introduction of attention-based modeling increased performance for particular natural-language processing tasks; thus, we utilize attention-based modeling in our work.

BERT Embeddings

For both convolutional neural networks (CNNs) and long short-term models (LSTMs) to work well, the architecture needs some way of converting words into floating-point numerical values that can be used within the deep-learning architecture. Thus, word embeddings were introduced as a way to transform text into numerical versions of those words [22]. Numerous pre-trained word embeddings have been introduced through research work over the years including GLOVE and word2vec embeddings [23, 24]. BERT has been introduced as a new form of pre-trained word embeddings which has increased performance for particular natural-language processing tasks; thus, we utilize BERT embeddings in our work [1].

CHAPTER III

PROBLEM STATEMENT

In this chapter, we formally define the problem of filtering high quality-reviews from an uncurated dataset. Additionally, we formally define the difference between review-based and sentence-based classification on the given problem.

3.1 Review-Based Classification

Formally, let $D = \{d_1, d_2, \ldots, d_n\}$ be a set of n review documents on which we will perform classification. Additionally, we will define a set $C = \{$Electronics, Kitchen & Dining, Home & Garden, Appliances, Travel, Health & Fitness, Baby & Kid, Outdoors, Pets$\}$ which is defined to be the set of review domain. For any given document $d \in D$, we define its review domain to be $c_d \in C$.

For each document $d \in D$, we define the review-based classification of high-quality or low-quality review to be defined as the set $Q = \{1, 0\}$ where 1 equates to a high-quality review and 0 equates to a low-quality review. The classification for a document d is defined as $q_d \in Q$. In particular, the problem of filtering high-quality reviews from an uncurated dataset involves taking a document $d \in D$ and assigning a value $q_d \in Q$ to the document d.

3.2 Sentence-Based Classification

Now, we expand the classification problem on sentence-based classification. For each document $d \in D$ defined in the previous subsection, let the sentences for a document d be defined $S_d = \{s_1, s_2, \ldots, s_m\}$ where m is the number of sentences in a particular document d. These sentences for each document will form the larger set S such that $S = \{S_{d_1} \cup S_{d_2} \cup \ldots \cup S_{d_n}\}$. Let the review domain of a sentence be defined to be c_s for any sentence $s \in S$ such that the following holds: $C' = \{c_{s_i} = c_d \mid \forall s_i \in S_d\}$.

For training, we define the sentence-based classification assignment of high-quality or low-

quality review based on the same method as review domain: $\{q_{s_i} = q_d \mid \forall s_i \in S_d\}$. As before, high-quality or low-quality review assigned is defined to be in the set $Q = \{1, 0\}$ where 1 equates to a high-quality review and 0 equates to a low-quality review. The classification for a sentence s is defined as $q_s \in Q$. In particular, the problem of filtering high-quality sentences from an uncurated dataset involves taking a sentence $s \in S$ and assigning a value $q_s \in Q$ to the sentence s.

3.3 Transforming Sentence-Based into Review-Based Classification

To transform the sentence-based classification problem into a review-based classification problem, we use Equation :

$$q_d = \begin{cases} 1 & \frac{1}{m} \sum_{s \in S_d} q_s \geq 0.5 \\ 0 & \frac{1}{m} \sum_{s \in S_d} q_s < 0.5 \end{cases} \quad \text{(Eq. 1)}$$

Thus, we have defined a functional mapping from sentence-based classification to review-based classification and we have a way of evaluating whether a review is of high-quality by its component parts.

CHAPTER IV

DATA COLLECTION

In the following chapter, we break the data collection process into two main sections: (i) First, we provide a bird's-eye overview of our curated and uncurated datasets for high-quality review filtering. (ii) Second, we describe the architecture through which we performed web scraping to attain the two datasets. Additionally, we show counts of both reviews and sentences, delineated by domain category, attained through our web scraping methods.

4.1 Dataset Overview

In this section, we provide an overview of the Wirecutter and Amazon datasets that we used to filter high-quality reviews from an uncurated dataset.

Wirecutter Reviews

As stated in the introduction, we obtain our curated high-quality review dataset from Wirecutter.com. The website contains different categories of products that include *Electronics*, *Home & Garden*, *Kitchen & Dining*, etc. For our work, we utilize nine different categories of products. Each of these categories contains articles which contain product reviews. On Wirecutter, we see two forms of reviews that we utilize in our classification task. First, Wirecutter contains their "top picks" reviews which contain information on products which they recommend and tend to have a highly positive sentiment. An example of a Wirecutter "top pick" review is shown in Figure 1.

Second, Wirecutter contains reviews that exist after they list their initial "top picks" reviews. Wirecutter experts label these reviews under the section "The competition." These reviews contain products which they have reviewed, but they view as lower quality than the products listed near the top of their articles. These reviews were incorporated into our dataset because they would offer a more negative sentiment than their "top picks" counterparts. An example of one of these reviews is shown in Figure 2.

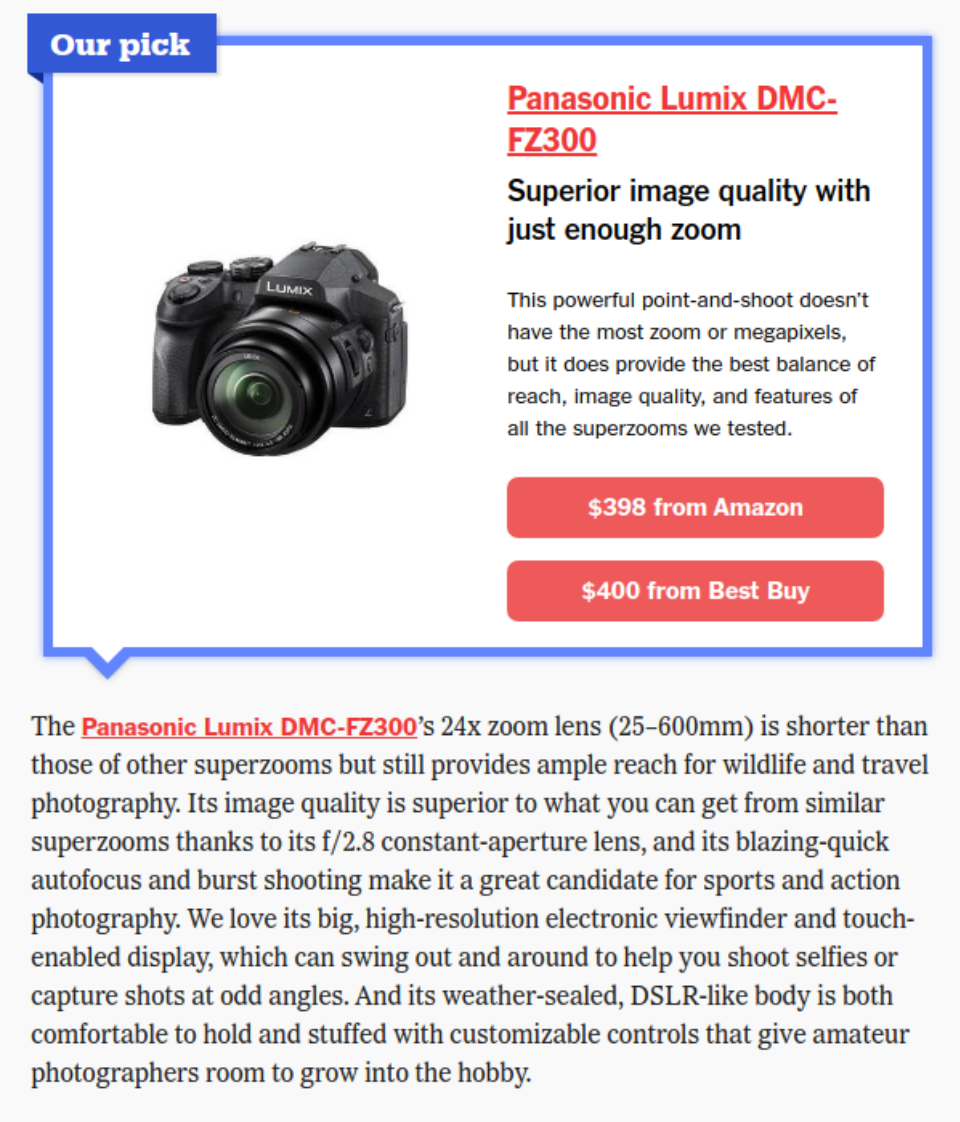

Figure 1: "Top pick" review under "The Best Superzoom Camera" Wirecutter article

Amazon Reviews

We obtain our uncurated review dataset from Amazon.com. Amazon contains crowd-sourced reviews from consumers who have purchased products off of the website. Users are asked to provide a title, verbal feedback, and a numerical "star" rating associated with the product. An example of an Amazon review is shown in Figure 3.

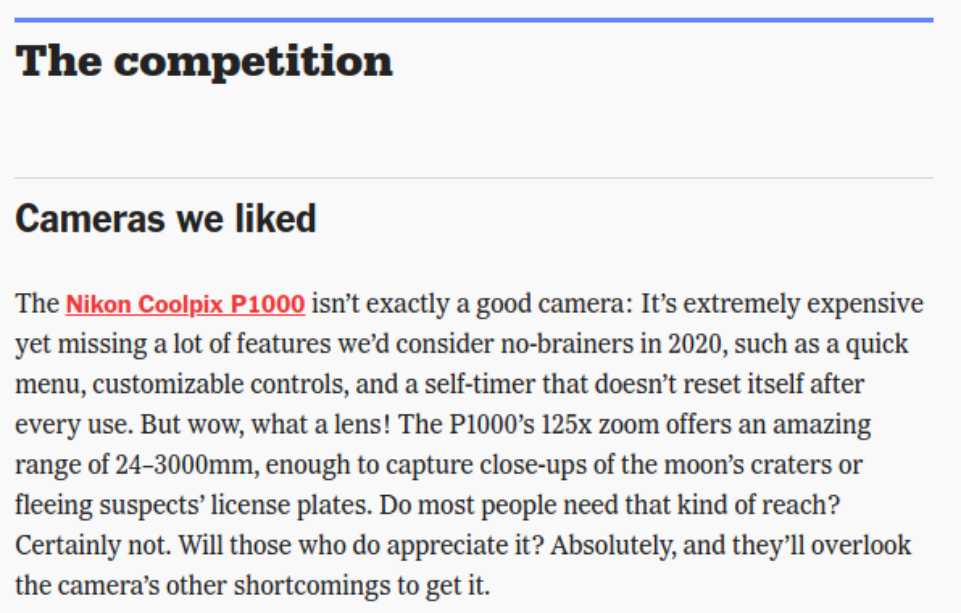

Figure 2: "Competition" review under "The Best Superzoom Camera" Wirecutter article

Figure 3: Amazon review corresponding to a product in Wirecutter article

4.2 Web Scraping

Prior to performing experiments on models and data sampling, we need to obtain the curated and uncurated data for both our Wirecutter and Amazon review datasets. While this data is online for anyone to view, the data is currently part of *html* documentation and is not in a format that is usable for our research. Thus, we need to use web scraping libraries to obtain the data from Wirecutter.com and Amazon.com. While we attempt to use the same web scraping program for both datasets, Amazon has certain checks to ensure its website is not being web scraped by bots.

Thus, we required two different web scraping programs discussed in more detail in the following section.

As stated in the previous section, we choose nine different categories of consumer products on which to perform experiments. To obtain the reviews from these products, we scraped each Wirecutter article on a category page using the architectural logic shown in Figure 4. We create a single CSV file containing data related to the product title, the review, the purchase link, and whether the purchase link associated with the product references an Amazon link.

For Amazon reviews, we obtain corresponding products to products that we find on Wirecutter. Because of the volume of reviews that correspond to some products on Amazon, we limit the number of reviews per product to 1000 reviews on Amazon. In addition to the review itself, we wish to obtain the title of each review, the user who wrote the review, and the stars associated with the review.

Because Amazon contains protections against bots and web scrapers, we are not able to use the same architecture as specified for Wirecutter reviews. Instead, we must use the Selenium web scraping library to further mimic the behavior of a human being during the web scraping process. This process is detailed in Figure 4.

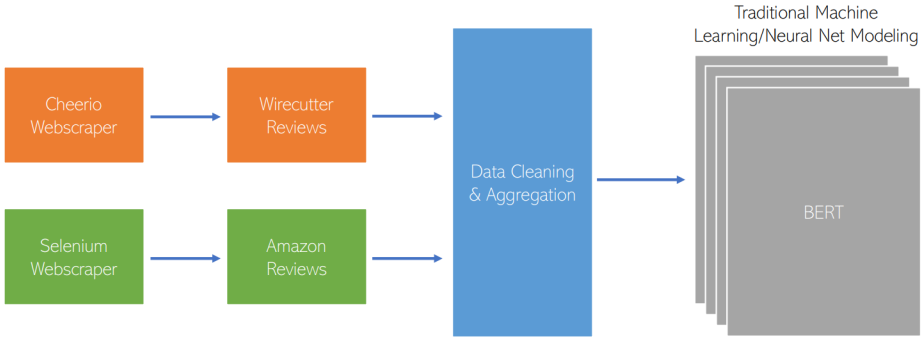

Figure 4: Web scraping architecture for Wirecutter and Amazon reviews

After performing data scraping, we create multiple CSV files each containing 1000 cor-

responding to products from the Wirecutter review dataset. Each line within each file contains the title, reviewer, review text, and stars of the review. The counts from scraping Amazon and Wirecutter reviews and separating the reviews into their respective domain categories are shown in Table 1. Additionally, we choose to break each review into its component sentences through the use of a Python tokenizer. The counts from breaking the Amazon and Wirecutter reviews into their component sentences delineated by domain category are shown in Table 2.

Table 1: Counts of Amazon and Wirecutter reviews delineated by category

Category	**# Wirecutter Reviews**	**# Amazon Reviews**
Electronics	2815	593,337
Home & Garden	2100	449,719
Kitchen & Dining	1244	324,159
Travel	418	84,030
Appliances	484	145,570
Health & Fitness	899	195,670
Baby & Kid	515	117,610
Outdoors	1029	112,110
Pets	293	131,050
Totals	**9797**	**2,153,255**

Table 2: Counts of Amazon and Wirecutter sentences delineated by category

Category	**# Wirecutter Sentences**	**# Amazon Sentences**
Electronics	7437	2,732,430
Home & Garden	6583	1,869,104
Kitchen & Dining	3451	1,250,705
Travel	1214	316,043
Appliances	1556	674,562
Health & Fitness	2620	803,129
Baby & Kid	1475	502,724
Outdoors	2671	440,048
Pets	942	657,115
Totals	**27,949**	**9,245,860**

By observing both Table 1 and Table 2, we can quickly see that there is a significantly higher number of uncurated, Amazon reviews than curated, high-quality, Wirecutter reviews. This makes sense, as there are going to be fewer high-quality reviews that exist online because the number of experts in a given field (in this case, reviewing products) is necessarily low based intrinsically on the definition of the word.

CHAPTER V

METHODS

In this chapter, we present the design of our high-quality review filtering approach. Namely, we present the process by which we progressed to developing more robust models for capturing the complex semantic and argumentative structure of high-quality reviews. By starting at baseline models and building to more robust, attention-based modeling, we are able to evaluate the performance increase compared to naïve modeling.

Our approach towards the difficult problem of classifying crowd-sourced reviews as high-quality reviews is four-pronged:

(i) First, we propose designing and implementing attention-based modeling to support deep semantic and argumentative structural complexity for high-quality review filtering.

(ii) Second, we propose designing and implementing other modern deep neural networks which will be able to attain performance which traditional machine learning models will not be able to attain. Through the process of comparing our attention-based modeling to other deep architectures, we hope to show the comparative improved performance of our attention-based modeling architecture.

(iii) Third, we propose using traditional machine learning models and other rudimentary models of minimal complexity to provide baseline performance from which we can compare both our deep architectural models and our attention-based models.

(iv) Finally, we propose performing experiments on our various models using different data-sampling strategies. By utilizing different data-sampling strategies, we hypothesize that we will be able to create a model that will generalize better across different review category domains.

5.1 Attention-Based Modeling

While LSTMs and CNNs have increased performance across the board from previously-used deep neural networks, deep bi-directional transformers (BERT) have recently been introduced by Google AI research and have shown astounding results compared to their RNN and CNN counterparts. While LSTMs and earlier embeddings such as ElMo focus on left-to-right or right-to-left training for embeddings, BERT embeddings focus on producing a true, bi-directional embedding, drastically increasing the results of traditional embeddings [1].

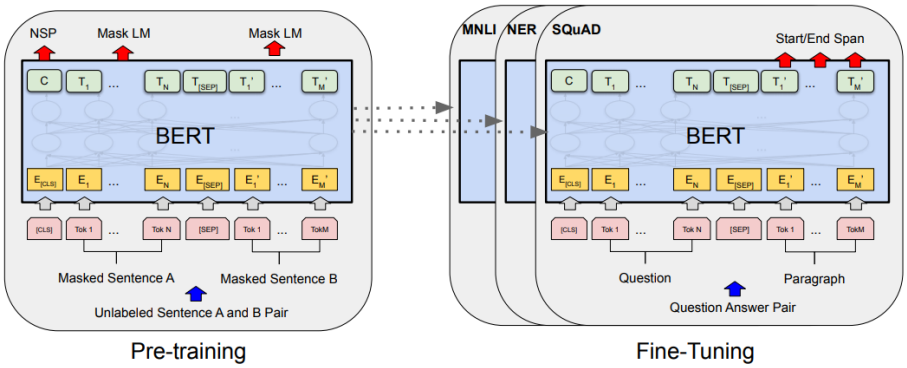

Figure 5: Figure from Devlin et al, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* visualizing BERT embeddings training process [1]

As stated in the BERT introductory paper on these embeddings, BERT embeddings leverage the transformer architecture first introduced in the paper "Attention is all you need" and shown in Figure 5 [1, 25]. By leveraging this architecture, the embeddings are able to better capture long-term dependencies and understand how sentences are related to one another, building semantic structure [25]. This process of capturing long-term dependency is further captured through the BERT tokenization process [1]. As stated in the Devlin, et al paper, "for a given token, its input representation is constructed by summing the corresponding token, segment, and position embeddings" [1]. This tokenization process can is visualized in Figure 6 [1].

Compared to other modern-day word embeddings, BERT has performed exceptionally well

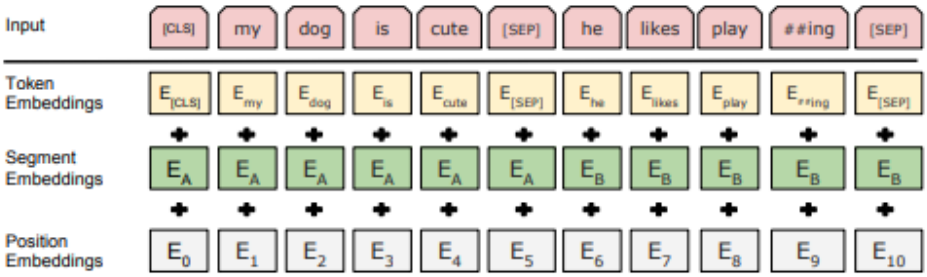

Figure 6: Figure from Devlin et al, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* visualizing various captured information in BERT tokenization process [1]

on language-modeling task including the General Language Understanding Evaluation benchmark (GLUE), the Stanford Question Answering Dataset (SQuAD), and the Situations With Adversarial Generations dataset (SWAG) [1]. Because we believe BERT embeddings to be one of the best representations of language modeling in current natural language processing research, we leverage these embeddings in our attention-based modeling architecture. The BERT paper proposes two sizes of embeddings of varying complexities based on the number of layers and "self-attention heads" [1]. These two sizes of embeddings are: $\text{BERT}_{\text{LARGE}}$ and $\text{BERT}_{\text{BASE}}$ [1]. Due to the complexity of our dataset and the need for more efficient training and classification, we choose to leverage the smaller $\text{BERT}_{\text{BASE}}$ embeddings for our modeling task.

5.2 Other Deep Architectures

We propose other advanced deep learning techniques to filter high-quality reviews from an uncurated dataset. Thus, we test performance using Deep Neural Networks, Convolutional Neural Networks, and Long Short-Term Memory Networks.

Deep Neural Networks

First, we propose the usage of deep neural networks as one of the models for our classification task. Deep neural networks are the basis for many other neural networks and have their foundation in the Perceptron, first introduced in a paper by Frank Roseblatt [26].

Later, this concept of a perceptron was built on to produce the modern deep neural network

that we know today. This network weights features based on vectorized input testing data, producing a classifier that is able to find correlation between classes that a traditional machine learning classifier would not be able to find. To create a floating-point vector form of our data for our deep neural network architecture, we leverage a TFIDF.

Convolutional Neural Networks

While deep neural networks may find some patterns within the classification of training data, they often struggle with long-term dependencies and understanding overall semantic structure. Thus, we introduce convolutional neural networks which use convolutions to develop short-term and long-term semantic structure in the classification process. Convolutions are performed in stages within the training process. After convolutions of the data are performed, pooling of this convolutional training data is performed. This process is repeated many times, ending in a standard deep neural network (or dense layer) that leads to the classification of data.

As with deep neural networks, we require a way to represent our text as numerical vector data. In the case of our convolutional neural networks, we leverage GloVe embeddings produced by Stanford Natural Language Processing research [23].

Long Short-Term Memory Networks

As with convolutional neural networks, we attempt to better capture argumentative and semantic structure in our classification model. Long short-term memory modeling has performed particularly well on capturing long-term dependency in natural language processing. LSTMs are a form of recurrent neural networks which feed a "memory" vector along with the original data vector through the network during training. This "memory" vector is trained along with the weights of the nodes within the network. As with convolutional neural networks, this architecture is repeated many times within the network, ending in a standard deep neural network (or dense layer) that leads to the classification of data. As with our convolutional neural network implementation, we represent our text as a floating-point vector using GloVe embeddings.

5.3 Baseline Classification Approaches

Prior to using more advanced deep learning models, we needed to develop a baseline performance metric by which to evaluate our later models. Additionally, we leverage different ways of converting text data into numeric data. Thus, in this section we propose the use of both TFIDF vectorization and Count vectorization. Additionally, we choose four very common baseline models to filter high-quality reviews from an uncurated dataset: Rocchio Algorithm Classification, and K-Means Feature Selection and Classification, Naïve Bayes, and Support Vector Machines.

TFIDF Vectorizers vs. Count Vectorizers

Traditional machine learning architectures require the use of numerical data in the training and classification of samples. Count vectorization is often the easiest form of producing numerical data from textual data. Count vectorization is formed by simply taking the number of times that a term appears in a given document or class (term-frequency) and using this value for training and classification.

For instance, Naïve Bayes classifiers often use term-frequency (TF) to determine which class a given document should be classified [27]. While this term-frequency metric can be helpful, there is still some essential data that is lost by simply counting the number of times a term appears in a given document. Namely, the "rareness" of a word or the amount of information that word conveys is left out of the equation by only using the term-frequency metric.

In 2004, Karen Spärck Jones introduced the concept of inverse document frequency (IDF) which formalized this notion of the "rareness" of a word [28]. According to Jones, "The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs" [28]. By combining this notion of term frequency with inverse document frequency, we are able to provide more information to our classification algorithm about the importance of a given word in classification. Term frequency-inverse document frequency is formalized in Eq. 2 where w is a given word, d is a given document, idf_w is the inverse document frequency of a word and $tf_{w,d}$ is the term frequency of the word w in the given document d. By providing this information, we are able to improve our performance on document classification. Thus, we implement TFIDF

vectorizer in our Support Vector Machine model to better capture features over a Naïve Bayes model.

$$\text{tf-idf}_{w,d} = idf_w \cdot tf_{w,d} \tag{Eq. 2}$$

Rocchio Algorithm

Our application of Rocchio classification involved the common technique of using centroids for decision boundaries. Training is performed by calculating the centroid for each class c within the larger dataset D [29]. The centriod for a given class c over each data point d within the dataset D "is computed as the vector average or center of mass of its members" given by the equation in Eq. 3 [29].

$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d) \tag{Eq. 3}$$

Then, classification is performed by determining which centroid has the greatest cosine similarity to the datapoint d [29]. Cosine similarity is defined in in the following way in Eq. 4 [30].

$$\cos(A, B) = \frac{A \cdot B}{||A||\ ||B||} = \frac{\sum_{i=0}^{n} A_i B_i}{\sqrt{\sum_{i=0}^{n} A_i^2} \sqrt{\sum_{i=0}^{n} B_i^2}} \tag{Eq. 4}$$

The process of classification using the Rocchio Algorithm of a data point d on a class c can be formalized using the equation in Eq. 5 [29].

$$c = \underset{c \in C}{\operatorname{argmax}} \cos(\vec{\mu}(c), \vec{v}(d)) \tag{Eq. 5}$$

K-Means Feature Selection and Classification

To perform classification using features that consisted of more than the words in the documents themselves, we use K-Means Feature Selection & Classification. The K-Means Feature Selection algorithm derives from methods used in the unsupervised machine learning method of K-Means Clustering [31]. We produced over twenty features from our datasets using features such

as Flesch Kincaid Readability, Sentiment Polarity, and counts of different parts of speech. By providing these features to the K-Means Feature Selection algorithm, the algorithm produced the top five and top ten features that best separate the two datasets. After producing this set of top five or top ten features that most separate the two datasets, we then run classification with our datasets leveraging these features and support vector machines.

Naïve Bayes

Naïve Bayes classifiers are frequently used as the baseline for many natural language processing classification tasks. They prove to be extremely efficient and perform classification on documents based on the probability of words appearing within different document classes [32]. To perform training use a Naïve Bayes classifier, we simply count each word that occurs in a given training document and develop a term-document matrix containing every word and every class of document [32]. Then, we attempt to estimate the probability of document d being in class c [32]. We write Eq. 6 to be the way that we define this probability [32].

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \tag{Eq. 6}$$

"where $P(t_k|c)$ is the conditional probability of term t_k occurring in a document class c" [32]. Because we cannot know the true conditional probabilities for terms and classes, we label the true conditional probability P and its estimation based on training $\hat{P}$ [32]. Classification in Naïve Bayes is performed using the function in Eq. 7 [32].

$$c = \underset{c \in C}{\operatorname{argmax}} \left[\log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k|c) \right] \tag{Eq. 7}$$

Support Vector Machines

We leverage Support Vector Machines (SVMs) in our traditional machine learning classification of high-quality reviews. The intuition behind SVMs is to maximize the boundary (or "margin") between the classes that we are classifying. To do this, we the Cost Function shown in Eq. 8 (including a regularization parameter λ (Eq. 8) which we wish to minimize. By minimizing

this cost function, we are maximizing the margin between the classification of two classes. We leverage SVMS as they often outperform traditional Logistic Regression by ignoring outliers.

$$\lambda||\vec{w}||^2 + \frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i(\vec{w}\cdot\vec{x_i} - b)) \quad \text{(Eq. 8)}$$

5.4 Data-Sampling Strategies

Another way in which we expanded our research and attempted to increase performance was through our data-sampling strategy. For data-sampling, we present two distinct methods for training our models: single-domain training and cross-domain training.

Single-Domain vs. Cross-Domain Training

By single-domain training, we mean training on a particular domain (or category) of Wirecutter and Amazon reviews and classifying over a large set of domains. Because choosing a single domain limits the number of high-quality product reviews we can use for training, we choose categories which have high counts of high-quality reviews. Thus, we use the Electronics dataset to perform training on single-domain classification.

By cross-domain training, we mean training using different domains (or categories) of Wirecutter and Amazon reviews combined together to form one large dataset. We then take this model that has been trained on a variety of domains and classify over categories that contain the training domains and over categories that do not contain the training domains. For example, in single-domain training, we train on Electronics samples from both Amazon and Wirecutter while with cross-domain training, we train on Electronics, Kitchen & Dining, and Home & Garden samples.

While in single-domain training, we are limited by the number of high-quality reviews that exist within our training domain, in cross-domain training, we have more than enough data for training. Thus, we are able to experiment with different domain combinations, determining which set of domains combined together produces the most generalized and accurate model. We explore how these combinations of domains affect our performance further in the results section of this

thesis.

Review-Based vs. Sentence-Based Classification

To start our research, we separated our data at the review-level and trained and tested our data using review-based classification. Through this methodology, we hope to achieve our original goal which entails classifying high-quality reviews from an uncurated dataset.

While directly classifying reviews as high or low-quality from an uncurated dataset seems to be the most intuitive way to solve our research problem, we hypothesize that using full-reviews for training causes overfitting in the model. Thus, we propose using a sentence-based method for classification which reduces overfitting and creates a more generalized model that works outside of the domains on which we trained and tested.

5.5 Evaluation Metric

To evaluate our performance on our different models, we utilize weighted F1-score to compare the performance on the sentence-base and review-based classification. An F-score is introduced in Nancy Chicor's papers on Evaluation Metrics [33]. Specifically, Chicor defines F-score as stated in Eq. 9 where P is precision and R is recall.

$$F = \frac{(\beta^2 + 1.0) \times P \times R}{\beta^2 \times P + R} \tag{Eq. 9}$$

In Eq. 9, β "is the relative importance given to recall over precision. If recall and precision are of equal weight, $\beta = 1.0$. For recall half as important as precision, $\beta = 0.5$. For recall twice as important as precision $\beta = 2.0$" [33]. In the case of our experiments, we take $\beta = 1.0$, where recall and precision are of equal importance. For weighted F1-score, we define a weighted version of a metric as one that uses F1-score metrics from each class and weights it based on the percentage that class contributes to the overall classified dataset. This weighted F1-score metric is shown in Eq. 10 where $C = \{c_1, c_2, \ldots, c_n\}$ is the set of all classes we classify, F_c is the F-1 score of the class c and W_c is the percentage of samples classified in class c from the entire dataset.

$$\text{weighted } F_1 = \sum_{c \in C} F_c \times W_c \qquad \text{(Eq. 10)}$$

We choose weighted F1-score as our metric for evaluation because of its ability to capture the importance of recall and precision while also accounting for differences in the size of different classifications.

CHAPTER VI

RESULTS

In this chapter, we show attention-based modeling as the best way to filter high-quality reviews from our uncurated dataset. To establish baseline models for comparison testing, we break our results into three major sections as follows: (i) First, we observe various features of the uncurated and curated datasets to understand the features that most differentiate the Wirecutter and the Amazon dataset. Thus, we will show features that we observed provided the greatest separation between the two datasets. Additionally, we will show how the common-sense understanding of these two datasets supports the differentiable features that we identify and gives us reason to believe that there is a specific structure to these high-quality reviews that we can model. (ii) Second, we provide precision, recall, and F1-score comparisons of our different modeling mechanisms performing classification of high-quality reviews on a uncurated dataset. (iii) Finally, we provide precision, recall, and F1-score comparisons of single-domain training for our highest performing models versus cross-domain training on these same models. Additionally, we provide a feature analysis comparing the highest weighted features used in DNN classification for single-domain training and cross-domain training. This analysis provides experimental evidence giving us further confidence that our model will generalize well outside of our specific experimental task.

6.1 Feature Analysis

In the following section, we observe the features that most differentiate the high-quality and low-quality reviewed datasets using a review and sentence-based approach.

Review-based Approach

After observing the counts of reviews and sentences retrieved through web scraping, we performed analysis to determine features that would separate the high-quality and low-quality reviews from each other. Initially, we looked at features revolving around counts of certain types of

punctuation and counts of parts-of-speech on average for both reviews and sentences.

Next, we look at more complex features that provide more insight into the semantic structure of the review. Namely, we will observe the readability and sentiment polarity scores of the datasets using different algorithms. Important features used to differentiate the two datasets can be found in Table 3. A full list of review-based features is found in Appendix A.

Table 3: Important review-based features for Amazon and Wirecutter datasets

Feature (Avg.)	**Wirecutter**	**Aamzon**
Word Length	4.91	4.46
Exclamation Count	0.00	0.51
Colon Count	0.23	0.10
Semicolon Count	0.08	0.03
Difficult Word Count	12.72	8.49
Sentiment Polarity	0.12	0.26
Dale Chall Readability	8.23	6.34
Automated Readability Index	13.10	6.98

Immediately, we notice features that would logically appear to differentiate the two dataset from one another. We would expect more intelligent, more knowledgeable writers to use longer words, less commonly used words (i.e. stop words), and more difficult words in their writing. Thus, we see that there are some rudimentary features that separate the two datasets.

As with some of the rudimentary features, we are able to see some separation between these two datasets. We observe, within the polarity feature, that Wirecutter reviews tend to be more neutral (closer to 0) while Amazon reviews tend to be more positive. Looking at Figure 7, we can see the difference between the reviews in the two datasets. Additionally, we observe from the readability features that, overall, Wirecutter reviews are harder to read because they are of a higher reading level than Amazon reviews. This makes sense, as we would expect higher-quality reviews to be written with language that is of a higher reading-level than lower-quality reviews. Looking at Figure 8, we see the defined difference between the readability of reviews in the two

datasets.

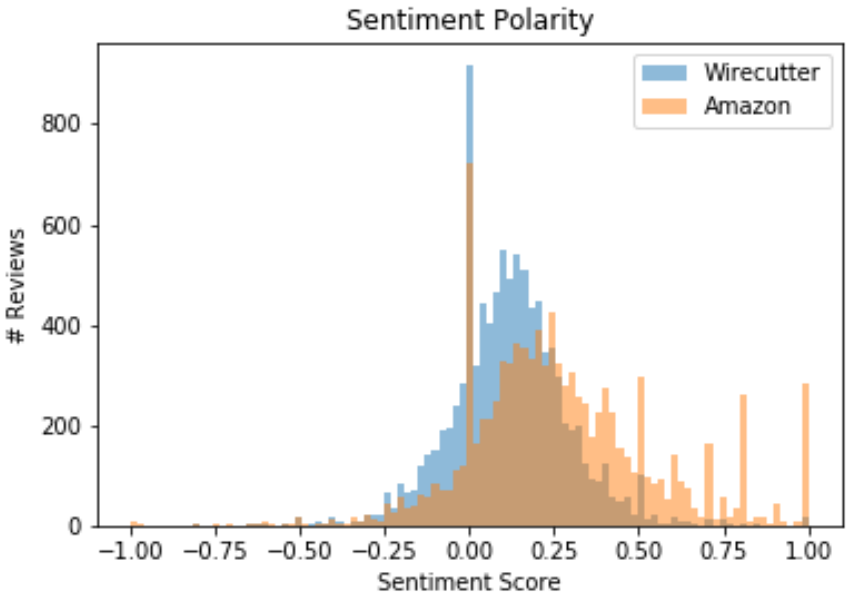

Figure 7: Sentiment polarity histogram of sentence-based Wirecutter and Aamzon datasets

Sentence-based Approach

As with review-based feature analysis, we observe the features that most differentiate the two datasets. Important features can be found in Table 4. A full list of sentence-based features can be found in Appendix A.

We observe that there is differentiation among the word length and difficult words used in sentences in the high-quality dataset. Additionally, as with review-based feature analysis, we observe readability differentiates the two datasets significantly; however, unlike review-based feature analysis, sentiment polarity does not significantly differentiate the two datasets using sentence-based feature analysis. This is observed by comparing review-based sentiment polarity in Figure 7 and sentence-based sentiment polarity in Figure 9.

By analysing both the sentence-based and review-based features of the two datasets, we observe that, simply by features alone, we are able to differentiate the two datasets. By basic

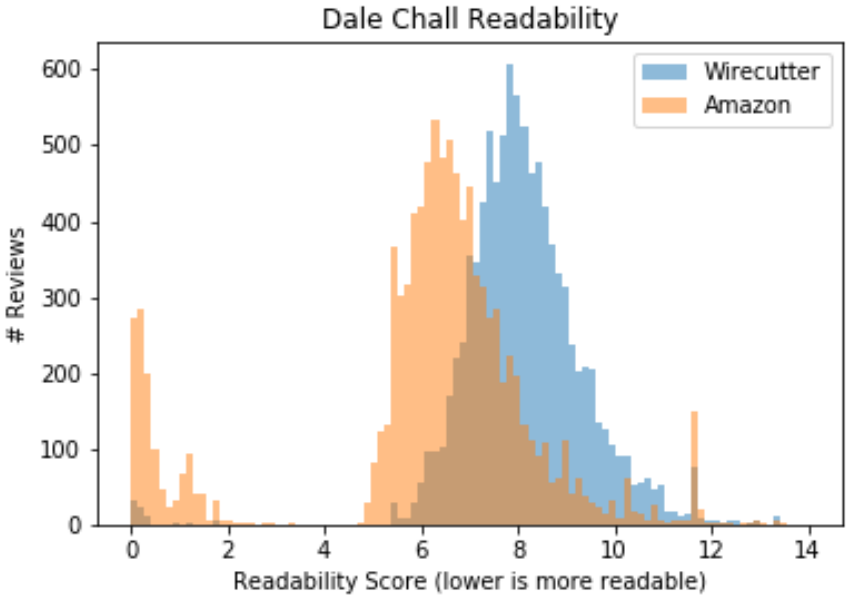

Figure 8: Readability histogram of Wirecutter and Aamzon review datasets

Table 4: Important sentence-based features for Amazon and Wirecutter datasets

Feature (Avg.)	**Wirecutter**	**Aamzon**
Character Count	127.49	79.77
Word Count	22.12	15.05
Word Length	4.86	4.49
Exclamation Count	0.00	0.12
Adjective Count	2.51	1.38
Difficult Word Count	4.76	2.23
Sentiment Polarity	0.10	0.19
Dale Chall Readability	8.10	6.10
Flesch Kincaid	9.81	6.03
Gunning Fog	11.85	8.39
Automated Readability Index	12.50	7.35

feature analysis, we can gain confidence that we will be able to build a model that will capture a high-quality review and differentiate between high-quality and low-quality datasets with high accuracy.

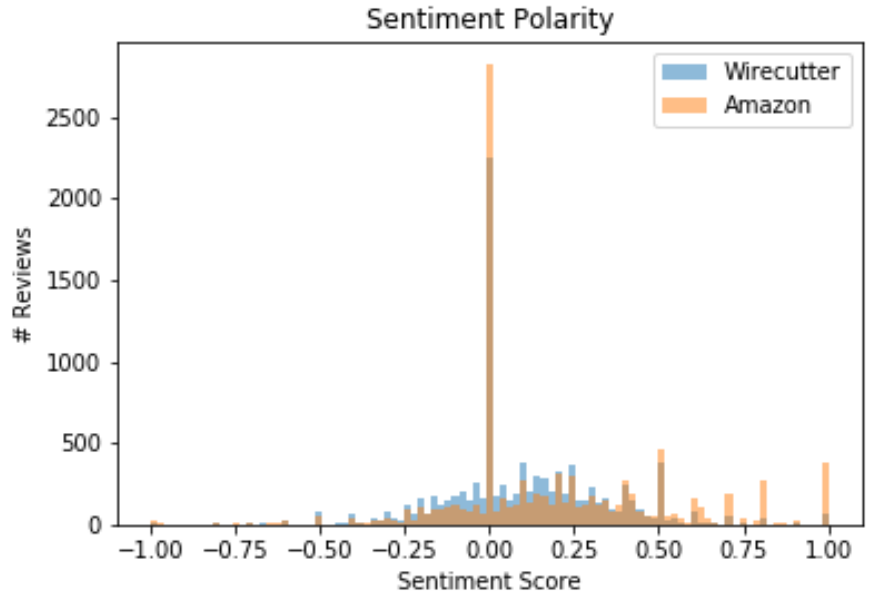

Figure 9: Sentiment polarity histogram of sentence-based Wirecutter and Aamzon datasets

6.2 Baseline Models

In this section, we observe the performance of our baseline models including: Rocchio Algorithm modeling, K-Means Feature Selection modeling, Naive Bayes modeling, and Linear Support Vector Machine modeling. Additionally, we observe the performance using both count vectorization and TFIDF vectorization for our training and classification of our textual data. We compare the weighted precision, recall, and f1-scores of these four models with our baseline models in Table 5. When we refer to the precision, recall, and f1-score metrics from our models, we are referring to the weighted versions of these metrics.

Already, we are seeing that we have incredibly high performance compared to our baseline by simply using more advanced machine learning methods. Additionally, though we tried to simply use some of our features, rather than words, in our k-means classification, we see that performance is less than stellar using this model. Most likely, we notice that the features that we used did not fully capture the relationship between a high-quality review and a low-quality review.

Now, we turn to see how our performance compares on sentence-based performance com-

Table 5: Baseline modeling review-based classification results

	Precision	**Recall**	**F1-Score**
Rocchio	0.62	0.56	**0.57**
K-Means (5 features)	0.77	0.61	0.53
K-Means (7 features)	0.76	0.59	0.49
K-Means (10 features)	0.72	0.59	0.52
NB	0.93	0.92	0.92
NB (TFIDF)	0.94	0.94	**0.94**
Linear SVM	0.96	0.96	0.96
Linear SVM (TFIDF)	0.97	0.97	**0.97**

pared to review-based performance. As before, we compare the weighted precision, recall, and f1-scores of the three machine learning models in Table 8; however, because K-means classification performed so poorly with review-based classification, we ignore it in sentence-based classification. Again, when we refer to the precision, recall, and f1-score metrics from our models, we are referring to the weighted versions of these metrics.

Table 6: Baseline modeling sentence-based classification results

	Precision	**Recall**	**F1-Score**
Rocchio	0.54	0.54	0.54
NB	0.86	0.86	**0.86**
NB (TFIDF)	0.85	0.84	0.83
Linear SVM	0.86	0.85	0.85
Linear SVM (TFIDF)	0.87	0.87	**0.87**

Surprisingly, basic Naive Bayes with a simple count vectorizer performs exceptionally well compared to other machine learning algorithms that we use as models. Still, Linear Support Vector Machines show the greatest performance separating high-quality review sentences from low-quality review sentences.

6.3 Deep Modeling

We now move to show results from performing deep learning classification on our data. In this section, we show results from deep learning classification using the following three models: Deep Neural Networks, Convolutional Neural Networks, and Long Short-Term Memory Networks. We compare the performance of our deep neural network models to those of our top-performing baseline models in Table 7.

Table 7: Deep modeling review-based classification results

	Precision	**Recall**	**F1-Score**
NB (TFIDF)	0.94	0.94	0.94
Linear SVM (TFIDF)	0.97	0.97	**0.97**
DNN	0.93	0.92	0.92
CNN	0.94	0.94	0.93
LSTM	0.94	0.94	**0.94**

We immediately observe that Support Vector Machines and Naive Bayes still have incredibly high performance on classification, even as compared to some deep architectures. Still, we note that, while we see high performance with Naive Bayes and Linear SVM models, these models will not generalize as well outside of our particular experiment architecture. For more generalized models, we turn to deep learning models that are better able to capture semantic and argumentative structure.

Observing the deep learning models as tested, we notice that a sequence model such as the LSTM network model actually performed extremely well compared to DNN and CNN modeling. Specifically, we note that all of LSTM, DNN, and CNN models have high performance on the review-based classification and filtering task.

For sentence-based classification tasks, we see much lower performance compared to review-based classification using deep modeling methods as with the traditional machine learning methods. Similar to review-based modeling, we see high performance on our Linear SVM modeling;

Table 8: Deep modeling sentence-based classification results

	Precision	**Recall**	**F1-Score**
NB (TFIDF)	0.85	0.84	0.83
Linear SVM (TFIDF)	0.87	0.87	**0.87**
DNN	0.86	0.86	0.86
CNN	0.89	0.88	**0.87**
LSTM	0.83	0.83	0.83

however, we do not see as high performance with LSTM modeling as we do in review-based modeling. Specifically with sentence-based modeling, we see that the CNN modeling outshines the other deep architectures as the best model for high-quality review filtering. This is somewhat expected as CNNs typically have high performance on classification tasks whereas LSTMs tend to have higher performance on question-answering and other sequence-based tasks.

6.4 Attention-Based Modeling

We now observe the performance of cutting-edge attention-based modeling with BERT embeddings on review-based and sentence-based high-quality review filtering. As before, we compare the precision, recall, and F1-scores of the highest performing baseline and deep models with our modern attention-based model through review-based classification in Table 9.

Table 9: Attention-based modeling (BERT) review-based classification results

	Precision	**Recall**	**F1-Score**
Linear SVM (TFIDF)	0.97	0.97	**0.97**
CNN	0.94	0.94	0.93
LSTM	0.94	0.94	0.94
BERT	0.99	0.99	**0.99**

Observing Table 9, we see that BERT modeling performs particularly well; however, this high of performance presents certain problems in terms of the generality of the model. While we recognize that the BERT model is the best model in terms of capturing the argumentative and

semantic structure of reviews, we require more training support to develop a generalized model for high-quality reviews. Still, compared to the highest performing baseline and deep learning models, we see outstanding performance from our attention-based model.

Now, we observe how our attention-based model with BERT embeddings performs through sentence-based classification for high-quality review filtering in Table 10.

Table 10: Attention-based modeling (BERT) sentence-based classification results

	Precision	**Recall**	**F1-Score**
Linear SVM (TFIDF)	0.87	0.87	**0.87**
CNN	0.89	0.88	**0.87**
BERT	0.96	0.96	**0.96**

As with review-based classification, we see that our attention-based modeling with BERT embeddings completely obliterates the other modeling architectures in sentence-based classification. Particularly, our BERT models show such high performance with sentence-based modeling that we say that they are very promising as a proof-of-concept in capturing high-quality semantics and high-quality argumentative structure.

6.5 Single-Domain Training vs. Cross-Domain Training

While single-domain training produced promising results, we attempt cross-domain training (that is, training using multiple domain categories) to increase performance and the generality of our models. Performing cross-domain training using review-based classification using our highest-performing baseline, machine learning, and deep learning models, we produce the following results shown in Table 11.

While cross-domain training offered very promising results, models (such as our BERT model) showed such high performance that they have a high potential for overfitting. Despite some of our models showing potential for overfitting, we see in Figure 10 that, overall, cross-domain training dramatically increased performance compared to single-domain training.

Table 11: Cross-domain training review-based classification results

	Precision	**Recall**	**F1-Score**
NB (TFIDF)	0.98	0.98	0.98
Linear SVM (TFIDF)	0.97	0.97	**0.97**
DNN	0.96	0.96	0.96
CNN	0.98	0.98	**0.98**
LSTM	0.92	0.91	0.91
BERT	1.00	1.00	**1.00**

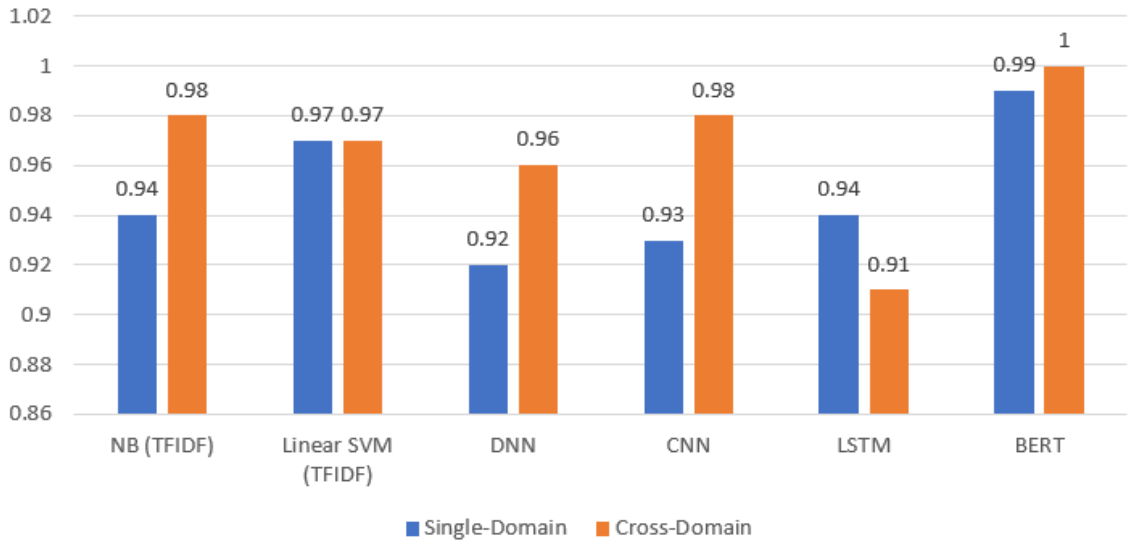

Figure 10: Comparison of review-based single-domain and cross-domain training

Looking at sentence-based modeling in Table 12, we see very similar results compared to our review-based modeling. Overall, cross-domain training in sentence-based modeling increased performance compared to its single-domain training counterpart. This difference can be seen in Figure 11 where cross-domain training outperformed single-domain training on each model.

To determine if cross-domain training increased our models' ability to generalize across different domains, we observed features from the training of deep learning models for both review-based and sentence-based classification as shown in Table 13. In both cases, we observe that features (or words) that carry more weight in the classification process appear to be more generalized

Table 12: Cross-domain training sentence-based classification results

	Precision	**Recall**	**F1-Score**
NB (TFIDF)	0.91	0.91	0.91
Linear SVM (TFIDF)	0.88	0.87	**0.87**
DNN	0.92	0.92	0.92
CNN	0.94	0.94	**0.94**
LSTM	0.94	0.94	0.94
BERT	0.98	0.98	**0.98**

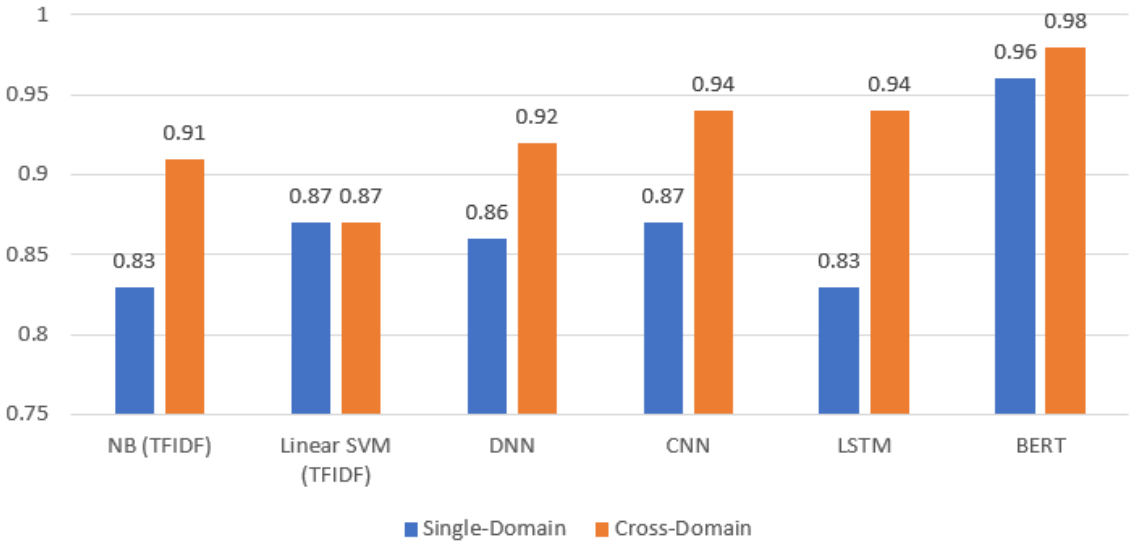

Figure 11: Comparison of sentence-based single-domain and cross-domain training

in cross-domain training than in single-domain training.

Table 13: DNN highest-weighted unique features training comparison

Training Domain	**Unique Features**
Electronics	bass, battery, camera, case, headphones, keyboard, phone, power, screen, sound, usb, wireless
Home & Garden	handle, light, works
Kitchen & Dining	coffee, cup, glass, handle, knife, model, pan, steel, water
Combined	better, easy, good, great, love, more, most, pick, price, quality, really sound, tested, time, very, works

6.6 Analysis of Results

We see that attention-based modeling coupled with BERT embeddings produces performance on filtering high-quality reviews that far surpasses any other deep learning or baseline model. Regardless of whether we performed data-sampling through review-based or sentence-based training and classification, attention-based modeling proved to best model the argumentative and semantic structure of high-quality reviews.

Additionally, cross-domain training appears to increase our model's ability to generalize across different categories. While it is difficult to understand specifically what features of a training dataset a deep learning model may be weighting highly, it gives us greater confidence that our model is generalizing well by observing the features that a deep neural network may be weighting highly through our TFIDF vectorizer.

In general, our results show that the complex argumentative and semantic structure of high-quality reviews can be modeled through mapping a curated high-quality review dataset onto an uncurated dataset. Additionally, we see evidence that our models generalize well outside of the particular classification task performed in this research but into other domain categories.

CHAPTER VII

CONCLUSION

By classifying high-quality reviews through different modeling techniques, our work is able to show that there is a structural and semantic difference between high-quality and low-quality reviews that can be captured through modern-day deep learning architectures. Specifically, we are able to best capture high-quality semantics through the usage of deep bi-directional transformer embeddings and a transformer (attention-based) architecture.

While some traditional machine learning-based modeling showed high performance at our classification task, the assumption is that these machine learning algorithms, which are based on vectors as representations of word-counts within the samples, will not generalize well outside of this specific task. In contrast, we expect that our modern-day deep learning architectures will be able to capture high-quality reviews on Amazon given enough training samples from a large domain space.

Additionally, we find that high-quality review structure is better captured when training on multiple domains as opposed to a single domain. This difference becomes even more apparent from viewing the features that the Deep Neural Network architecture was giving the highest weight through its classification process.

In general our experimental conclusions can be summarized through the following statements:

(i) We show that attention-based modeling can capture complex semantic and rhetorical structure of a high-quality review dataset to produce a classifier that can identify high-quality reviews from an uncurated dataset.

(ii) We show that our model can generalize across different domain categories despite training on a limited number of domain categories.

(iii) We show that our modeling captures a generalized structure through the analysis of features heavily weighted by the classifier

6.1 Open Questions

While our research captured very interesting results and shows promise for future work, extending previous work surrounding fake user review research, we present open questions that require further investigation as research into this area continues. These questions are formulated as follows:

(i) Though we hypothesize our data can be generalized outside of our specific classification task, does our modeling in fact extend beyond the task on which our models were trained?

(ii) While cross-domain training proved to increase performance on our particular classification task, does this cross-domain training produce over-fitting in our models?

(iii) Can more fine-tuning and hyper-parameter-tuning on our transformer-based model produce higher performance while additionally keeping our model generalized to perform its intended task?

(iv) How do we extend this research to aggregate high-quality reviews on consumer review platforms such as Amazon and Yelp to increase consumer awareness into the overall sentiment and usability of consumer products?

(v) How do we attain more reviews that parallel consumer reviews from Amazon, Yelp, etc. while also capturing the "high-quality" aspects presented in this paper?

6.2 Closing Thoughts

While modeling high-quality consumer product review structure through transformer networks coupled with BERT embeddings proved extremely promising in concept, there is still a large amount of research that needs to be done into this area. More research needs to be conducted

into developing more robust deep models that better map high-quality expert reviews onto high-quality consumer reviews. Our research presents a proof-of-concept into this area that we hope will be extended into the application of this type of work onto production-level consumer product marketplaces.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," Oct. 2018.

[2] "Retail e-commerce sales in the united states from 2017 to 2024." https://www.statista.com/statistics/272391/us-retail-e-commerce-sales-forecast/, 2020.

[3] B. Skiera, T. Hennig-Thurau, E. Malthouse, C. Friege, S. Gensler, L. Lobschat, and A. Rangaswamy, "The impact of new media on customer relationships," *Journal of Service Research*, vol. 26, 08 2010.

[4] N. Jindal and B. Liu, "Opinion spam and analysis," Feb. 2008.

[5] M. Hu and B. Liu, "Mining and summarizing customer reviews," *KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 168–177, Aug. 2004.

[6] C. Castillo, Carlos, D. Donato, Debora, L. Becchetti, Luca, P. Boldi, Paolo, Leonardi, S. Fanti, Santini, Massimo, Vigna, and Sebastiano, "A reference collection for web spam," *SIGIR Forum*, vol. 40, pp. 11–, Dec. 2006.

[7] A. Mukherjee, B. Liu, and N. Glance, "Spotting fake reviewer groups in consumer reviews," *WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web*, May 2012.

[8] S. Mudambi and D. Schuff, "What makes a helpful online review? a study of customer reviews on amazon.com.," *MIS Quarterly*, vol. 34, pp. 185–200, 03 2010.

[9] "About us." https://thewirecutter.com/about/, 2020.

[10] K. Dave, S. Lawrence, and D. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," *Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews*, vol. 775152, Nov. 2003.

[11] Q. Ye, J. Zhang, and R. Law, "Sentiment classification of online reviews to travel destinations by supervised machine learning approaches," *Expert Syst. Appl.*, vol. 36, pp. 6527–6535, Apr. 2009.

[12] C. Castillo, Carlos, D. Donato, Debora, A. Gionis, Aristides, Murdock, Vanessa, Silvestri, and Fabrizio, "Know your neighbors: Web spam detection using the web topology," July 2007.

[13] D. Fetterly, M. Manasse, M. Najork, and L. Avenida, "Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages," 07 2004.

[14] V. Krishnan and R. Raj, "Web spam detection with anti-trust rank.," pp. 37–40, 01 2006.

[15] M. Hu and B. Liu, "Mining and summarizing customer reviews," *KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 168–177, Aug. 2004.

[16] H.-J. Min and J. Park, "Identifying helpful reviews based on customer's mentions about experiences," *Expert Systems with Applications*, vol. 39, p. 11830–11838, Nov. 2012.

[17] A. Ghose and P. Ipeirotis, "Designing novel review ranking systems: Predicting usefulness and impact of reviews," *ACM International Conference Proceeding Series*, vol. 258, Aug. 2007.

[18] S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti, "Automatically assessing review helpfulness.," *EMNLP*, pp. 423–430, Jan. 2006.

[19] J. Liu, Y. Cao, C.-Y. Lin, Y. Huang, and M. Zhou, "Low-quality product review detection in opinion summarization.," pp. 334–342, Jan. 2007.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, Dec. 1997.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," June 2017.

[22] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, Oct. 2013.

[23] C. D. M. Jeffrey Pennington, Richard Socher, "Glove: Global vectors for word representation." https://nlp.stanford.edu/projects/glove/, 2020.

[24] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method," Feb. 2014.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 06 2017.

[26] F. F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65 6, pp. 386–408, 1958.

[27] "Term frequency and weighting." https://nlp.stanford.edu/IR-book/html/htmledition/term-frequency-and-weighting-1.html, 2008.

[28] K. Jones, "A statistical interpretation of term specificity in retrieval," *Journal of Documentation*, vol. 60, pp. 493–502, Jan. 2004.

[29] "Rocchio classification." https://nlp.stanford.edu/IR-book/html/htmledition/rocchio-classification-1.html, 2008.

[30] S. Prabhakaran, "Cosine similarity – understanding the math and how it works (with python codes)." https://www.machinelearningplus.com/nlp/cosine-similarity/, Oct. 2018.

[31] C. Boutsidis, M. Mahoney, and P. Drineas, "Unsupervised feature selection for the k-means clustering problem," *Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference*, pp. 153–161, Jan. 2009.

[32] "Naive bayes text classification." https://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html, 2008.

[33] N. Chinchor, "Muc-3 evaluation metrics," pp. 17–24, 01 1991.

APPENDIX A

DATASET FEATURE LIST

Table 14: All review-based features for Amazon and Wirecutter

Feature (Avg.)	**Wirecutter Reviews**	**Aamzon Reviews**
Character Count	366.04	340.91
Word Count	63.17	63.87
Word Length	4.91	4.46
Sentence Count	2.82	4.01
Syllables	89.17	84.13
Exclamation Count	0.00	0.51
Colon Count	0.23	0.10
Semicolon Count	0.08	0.03
Period Count	2.97	4.43
Comma Count	3.55	1.97
Stop Words Count	23.99	26.33
Noun Count	20.44	14.55
Verb Count	9.74	12.39
Adjective Count	7.24	5.85
Adverb Count	3.61	4.91
Proper Nouns Count	7.20	2.05
Difficult Word Count	12.72	8.49
Sentiment Polarity	0.12	0.26
Flesch Reading	62.51	79.37
Dale Chall Readability	8.23	6.34
Smog Index	5.37	4.70
Flesch Kincaid	10.21	5.83
Coleman Liau Index	10.07	5.68
Linsear Write	13.04	7.32
Gunning Fog	12.09	8.16
Automated Readability Index	13.10	6.98

Table 15: All sentence-based features for Amazon and Wirecutter

Feature (Avg.)	**Wirecutter Sentences**	**Aamzon Sentences**
Character Count	127.49	79.77
Word Count	22.12	15.05
Word Length	4.86	4.49
Sentence Count	1.00	1.00
Syllables	31.21	19.86
Exclamation Count	0.00	0.12
Colon Count	0.09	0.03
Semicolon Count	0.03	0.01
Period Count	1.04	1.06
Comma Count	1.24	0.49
Stop Words Count	8.41	6.19
Noun Count	7.18	3.44
Verb Count	3.41	2.89
Adjective Count	2.51	1.38
Adverb Count	1.26	1.15
Proper Nouns Count	2.54	0.45
Difficult Word Count	4.76	2.23
Sentiment Polarity	0.10	0.19
Flesch Reading	64.05	78.45
Dale Chall Readability	8.10	6.10
Smog Index	0.00	0.00
Flesch Kincaid	9.81	6.03
Coleman Liau Index	9.68	5.36
Linsear Write	12.62	7.73
Gunning Fog	11.85	8.39
Automated Readability Index	12.50	7.35